

## A Multi-Target Tracking Technique for Mobile Robots using a Laser Range Scanner

Polychronis Kondaxakis, Stathis Kasderidis, Panos Trahanias  
Institute of Computer Science  
Foundation for Research and Technology – Hellas (FORTH)  
E-mail: {konda, stathis, trahania}@ics.forth.gr

**Abstract**— A major issue in the field of mobile robotics today is the detection and tracking of moving objects (DATMO) from a moving observer. In dynamic and highly populated environments, this problem presents a complex and computationally demanding task. It can be divided in sub-problems such as robot's relative motion compensation, feature extraction, measurement clustering, data association and targets' state vector estimation. In this paper we present an innovative approach that addresses all these issues exploiting various probabilistic and deterministic techniques. The algorithm utilizes real laser-scanner data to dynamically extract moving objects from their background environment, using a time-fading grid map method, and tracks the identified targets employing a Joint Probabilistic Data Association with Interacting Multiple Model (JPDA-IMM) algorithm. The resulting technique presents a computationally efficient approach to already existing target-tracking research for real time application scenarios.

### I. INTRODUCTION

FOR mobile robot applications that require to operate in dynamic and highly populated environments (e.g. museums, hospitals, etc.), it is imperative to identify and track the trajectories of moving objects. Predicting environmental dynamics, using target tracking techniques, provides important information for navigational algorithms and improves robot responses to dynamic changes. Many previous approaches, which utilize 2D laser range scanner measurements, have dealt with the DATMO problem assuming that previous knowledge on the background and the targets is provided to the algorithm. In [1], a system for tracking pedestrians is described, using multiple stationary and sparsely-placed laser range scanners. A background is obtained beforehand and subtracted from each measurement frame in order to extract possible moving objects. A Kalman filter (KF) is implemented to track the extracted targets. In a different trail of thought, the authors in [2], [3] develop a feature detection system for real-time identification of geometric shapes such as lines, circles and legs from laser data and apply a KF for tracking. A similar but more sophisticated method can be found in [4]. Geometric profiles

are created on raw laser measurements, based on fourteen conditions. A classifier trained on these profiles, searches for specified patterns that can be categorized as legs.

Grid-map approaches that employ laser data and particle filters for moving-object tracking can be found in [5], [6]. In both papers, a similar method is obtained that utilizes a series of occupancy grids for extracting moving objects in a dynamic environment. The authors introduce the sample-based Joint Probabilistic Data Association Filter (SJPDF), which is an adapted version of JPDA algorithm suited for particle filter applications. Although both publications demonstrate effective usage of this technique, maintaining two consecutive probabilistic grid maps of robot's space and additionally generating one map for every detected feature each time, is a process and memory expensive architecture. A grid map-free approach is presented in [7]. In this method, laser scanner measurements are either connected together in a closed polygon shape representing the free space, or as point samples of object surfaces. If such points are detected inside any previous polygon of free space, they are marked as moving object points. Odometry is employed to compensate the robot's relative movement. Furthermore, statistical hypotheses are propagated using a gradient ascent method to track moving targets. The technique is tested with only one available moving target in space, thus its efficiency in multi-target tracking scenarios is unknown. Finally, a method for outdoor mobile robot applications is presented in [8]. It combines Simultaneous Localization and Mapping (SLAM) with DATMO to detect, categorize and track multiple moving targets in urban environments.

This paper presents a new approach to DATMO problem framework. It blends the hybrid JPDA-IMM data association and tracking algorithm with an innovative occupancy-grid based approach for moving object detection. To extract and verify moving targets from stationary background, the proposed model follows a number of consecutive steps. Initially, a single occupancy-grid map is created. Every individual grid-cell accumulates a counter, which increases linearly when a laser measurement falls inside its occupancy area. Grid-cells with values above a dynamic threshold level are selected as stationary background. Moreover, all cells containing values above zero decrease linearly with time. A certain background is obtained and subtracted (utilizing

---

This work is supported by the FP6-Information Society Technologies program, European Commission INDIGO research project (Contract No. IST-045388).

covariance information) from every laser data frame leaving only the measurements that represent possible moving targets. The remaining measurements are clustered into groups and a centre of gravity is assigned to each one. Finally, the JPDA-IMM initiates and allocates tracks to clusters that exceed a certain velocity level. The resulting technique provides a processing and memory-efficient architecture and handles grid-map related problems such as map-size and transformation resolution.

## II. STATIONARY BACKGROUND ATTENUATION

### A. Laser Data Collection

This research was conducted using a SICK LMS200 Laser Range Scanner adjusted onboard a PIONEER 3-AT mobile robot platform. The scanner has an angular resolution of  $0.5^\circ$  and a span of  $180^\circ$  thus delivering 361 range readings per frame. Optimally, it can provide 25 frames per second at a maximum range of  $60m$  with a resolution of  $\pm 5cm$ . Due to local wireless network limitations the scanner delivers about five measurement frames per second. Moreover, as a result of the indoor operational specifications in our experiments, the laser was adjusted to detect obstacles at a maximum of  $9.6m$  distance. Measurements above  $9.6m$  were ignored as outliers. We represent a measurement frame at time index  $k$  by a vector  $\mathbf{z}_{laser}(k) = \{z_1(k), z_2(k), \dots, z_{361}(k)\}$ , where  $z_i(k)$  with  $l=1\dots 361$ , is a vector composed by distance and bearing readings,  $z_i(k) = [r_i(k), \varphi_i(k)]^T$ .

### B. Mobile Robot's Relative Movement Compensation

Currently, there are large numbers of available publications in the field of real-time multi-object tracking techniques. However, the existing research publications decrease dramatically if the observer is also moving. In mobile robotic applications, relative movement compensation could be achieved by either synchronizing pure odometry readings with laser scanner measurements [7] or combining a SLAM algorithm with a DATMO system in a single framework, thus estimating at the same time both moving objects in the environment and robot's trajectory [8].

This research utilizes an asynchronous distributed architecture where a number of subsystems (Localization module, Path-Planning module, DATMO module etc.) operate in parallel to obtain, upon request, the robot's higher functionalities. It is well known that a vehicle with skid-steer drive configuration such as the PIONEER 3-AT is prompt to odometric errors. For that reason, the DATMO does not solidly utilize robot's odometry. It rather exploits state and error covariance estimations, acquired from the localization algorithm [9], in order to compensate relative movement. The estimated state vector is composed by position  $(x, y)$  and orientation  $\theta$  components,  $\hat{\mathbf{x}}_R(k|k) = [\hat{x}_R(k|k), \hat{y}_R(k|k), \hat{\theta}_R(k|k)]^T$ . Its associated estimated state error covariance matrix is given

by  $\mathbf{P}_R(k|k)$ . It has been assumed that an estimated state vector is acquired simultaneously with every laser measurement frame,  $\mathbf{z}_{laser}(k)$ . Therefore, at discrete time index  $k$ , the robot's state and laser measurement readings are bundled into a duple of vectors  $\langle \mathbf{z}_{laser}(k), \{\hat{\mathbf{x}}_R(k|k), \mathbf{P}_R(k|k)\} \rangle$ . The algorithm is configured to collect samples at approximately constant periodicity. The scan points are transformed in robot's coordinate frame as follows:

$$\mathbf{z}_i^*(k) = \begin{bmatrix} x_i^*(k) \\ y_i^*(k) \end{bmatrix} = \begin{bmatrix} r_i(k) \cos(\varphi_i(k) + \hat{\theta}_R(k|k)) + \hat{x}_R(k|k) \\ r_i(k) \sin(\varphi_i(k) + \hat{\theta}_R(k|k)) + \hat{y}_R(k|k) \end{bmatrix} \quad (1)$$

with  $l=1\dots 361$  and  $\mathbf{z}_{laser}^*(k) = \{z_1^*(k), z_2^*(k), \dots, z_{361}^*(k)\}$ .

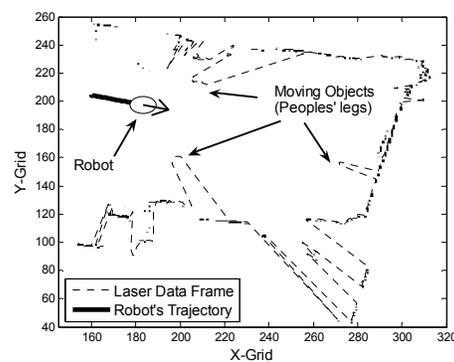


Fig. 1. Occupancy grid map for obtaining stationary background information.

### C. Occupancy Grid Map

In DATMO research, a very popular method for isolating moving objects in dynamic environments, utilizes multiple occupancy grid maps [5], [6], [8]. Our novel approach employs a single accumulative occupancy grid for stationary feature extraction. This grid-map is dynamically updated for every laser measurement frame,  $\mathbf{z}_{laser}^*(k)$ . When new laser data become locally available to the robot, at a discrete time instant  $k$  with  $k \in \mathbb{N}$ , they are translated in egocentric grid-map coordinates and assigned to cells. Every individual cell in the map maintains a counter, which increases linearly when a laser measurement vector,  $z_i^*(k)$ , is allocated inside its occupancy area. Assuming that cells with counts above a certain dynamic threshold level (e.g.  $3 \leftrightarrow 20$  depending on robot's relative movement) are associated with stationary background objects, the algorithm extracts background grayscale-grids as shown in Fig. 1. In the same figure, the solid line represents the robot's traveled trajectory and the dashed line represents the laser measurement frame vector  $\mathbf{z}_{laser}^*(k)$  as a line graph. Each square cell in the grid-map has a  $5$  by  $5cm$  resolution. Furthermore, an innovative property of the constructed grid-map is that as time progresses, the counts inside the cells decrease linearly (one count per iteration)

until they reach a zero value. In other words, if there is no positive laser measurement hits, grid counts tend to decrease at a constant rate. This technique dynamically updates the stationary background knowledge, limiting at the same time the required memory usage and processing power-consumption by the grid-map algorithm. Finally, the obtained background points are transformed back to robot's local coordinate frame and supplied to the tracking algorithm for further processing. They are represented by vector  $\mathbf{V}_{background}(k) = \{\mathbf{p}_1(k), \dots, \mathbf{p}_m(k)\}$ ; where  $\mathbf{p}_o(k)$  with  $o=1..m$ , is also a vector composed by  $x$  and  $y$ -stationary point components,  $\mathbf{p}_o(k) = [x_o(k), y_o(k)]^T$ . Fig 2. demonstrates the extracted background points in robot's local coordinate axis.

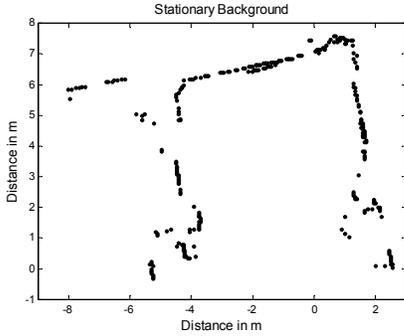


Fig. 2. Extracted background in robot's local coordinate frame.

#### D. Moving Target Detection

To facilitate the detection of possible moving objects' in robot's dynamic environment, the method deducts the background vector  $\mathbf{V}_{background}(k)$  from a laser measurement frame  $\mathbf{Z}_{laser}^*(k)$ , both obtained for time index  $k$ . Three major issues regarding the correct alignment of background and laser measurements are presented as follows: (i) The positional error and thus the relative error covariance estimates, increase as the mobile robot moves in space. (ii) The time required by this technique to readjust the dynamic background, increases as the robot moves. (iii) The laser measurement frame and dynamic background vectors are composed by different number of vector-points; hence one-to-one linear subtraction becomes prohibited. To resolve these issues, the proposed algorithm employs the following innovations.

Firstly, a dynamic threshold was introduced. It filters out possible moving targets from stationary objects by altering its condition according to robot's state. If the robot is on the move, the threshold level decreases linearly with time until it reaches a minimum value (e.g. 3). Therefore, the dynamic background vector ( $\mathbf{V}_{background}(k)$ ), becomes more susceptible to rapid environmental changes. The opposite happens, when the mobile robot remains stationary. In that case, the

threshold level increases linearly with time until it reaches a maximum value (e.g. 20), thus rendering the background vector less sensitive to changes.

To counteract additional translation and rotation point-set differences between vectors  $\mathbf{Z}_{laser}^*(k)$  and  $\mathbf{V}_{background}(k)$ , an Iterative Closest Point (ICP) algorithm is employed. ICP is attractive because of its simplicity and its performance. Given two point sets  $A$  and  $B$  in  $\mathbb{R}^2$  it minimizes a geometric distance function  $f(A+T, B)$ , over all translations  $T$ . The algorithm starts with an arbitrary translation that aligns  $A$  to  $B$ , and then repeatedly performs local improvements that keep re-aligning  $A$  to  $B$ , while decreasing the distance function until a convergence is reached. Although the initial estimate does need to be reasonably good, the algorithm converges relatively quickly. A closed-form solution to the ICP algorithm can be found in [10]. However, our experimental results indicate that in many cases, discarding the ICP algorithm and running only with the dynamic threshold, provides the same or even better results. This is due to ICP's inefficiency on large transformational changes.

Finally, the tracking algorithm removes the measurement points from  $\mathbf{Z}_{laser}^*(k)$  that are associated with vector points from  $\mathbf{V}_{background}(k)$  utilizing the provided robot's error covariance matrix estimation  $\mathbf{P}_R(k|k)$  and the laser-scanner's uncertainty model. Assuming the laser-scanner noise model in polar coordinates as [9]:

$$\mathbf{R}_{l_{polar}}(k) = \begin{bmatrix} k_{\varphi} \varphi_l(k) & 0 \\ 0 & k_{\rho_0} + k_{\rho_l} r_l(k) \end{bmatrix} \quad (2)$$

where  $k_{\varphi}, k_{\rho_0}, k_{\rho_l}$  are parameters related to the range finder angular error, range error independent of distance, and distance-dependent range error, respectively. The transformation of Polar to Cartesian coordinates is:

$$\mathbf{R}_{l_{cart}}(k) = \nabla \mathbf{C}(k) \mathbf{R}_{l_{polar}}(k) \nabla \mathbf{C}^T(k) \quad (3)$$

where  $\nabla \mathbf{C}(k)$  is the Jacobean of  $\mathbf{z}_l^*(k)$  with respect to  $[r_l(k), \varphi_l(k)]$ . As it is evident from equations (2) and (3), the covariance of the sources of uncertainty and noise for the available measurement vector is a time-varying matrix. The values of the elements of this matrix depend on the measurement distance and bearing components. Averaging across all possible values of  $\varphi_l(k)$  and utilizing sensors' maximum possible distance measurement ( $r_{l_{max}} = 9.6m$ ), the noise covariance matrix  $\mathbf{R}_{l_{cart}}(k)$  becomes:

$$\bar{\mathbf{R}}_{l_{cart}} = \begin{bmatrix} \frac{r_{l_{max}}^2 (k_{\varphi} \bar{\varphi}_l) + (k_{\rho_0} + k_{\rho_l} r_{l_{max}})}{2} & 0 \\ 0 & \frac{r_{l_{max}}^2 (k_{\varphi} \bar{\varphi}_l) + (k_{\rho_0} + k_{\rho_l} r_{l_{max}})}{2} \end{bmatrix} \quad (4)$$

Equation (4) provides an averaged time-invariant expression

of (3) and will be used for a track initiation and maintenance process, in order to improve the algorithm's calibration characteristics.

Finally, the overall error that describes the uncertainty of background point vectors is given by:

$$\mathbf{P}_{lback}(k) = \mathbf{H}_R \mathbf{P}_R(k|k) \mathbf{H}_R^T + \mathbf{R}_{lcart}(k) \text{ where } \mathbf{H}_R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (5)$$

A measurement point described by vector  $\mathbf{z}_i^*(k)$  is removed if and only if the Mahalanobis distance to any of the vector points included in  $\mathbf{V}_{background}(k)$ , is below a certain threshold  $\lambda$  resulting in:

$$\sqrt{[\mathbf{z}_i^*(k) - \boldsymbol{\rho}_{1,2,\dots,m}(k)]^T [\mathbf{P}_{lback}(k)]^{-1} [\mathbf{z}_i^*(k) - \boldsymbol{\rho}_{1,2,\dots,m}(k)]} \leq \lambda \quad (6)$$

Therefore, the remaining moving object points are represented as  $\mathbf{Z}_{moving}^*(k) = \{\mathbf{z}_1^*(k), \mathbf{z}_2^*(k), \dots, \mathbf{z}_s^*(k)\}$  with  $\mathbf{Z}_{moving}^*(k) \in \mathbf{Z}_{laser}^*(k)$  and  $s \leq 361$ . The effectiveness of the proposed method is clearly demonstrated in Fig 3. where the three moving objects appeared in Fig 1., have been detected.

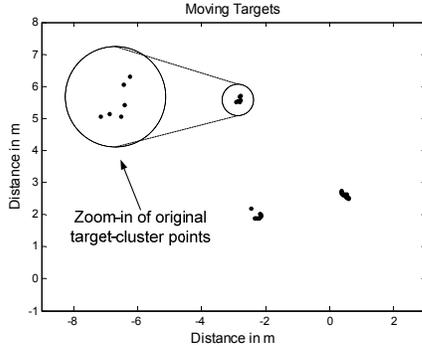


Fig. 3. Three moving objects detected

### III. LASER DATA CLUSTERING

To initiate the target tracking procedure, the segmentation of the raw data points included in vector set  $\mathbf{Z}_{moving}^*(k)$  provides clusters of scan points that are grouped together. Each cluster represents a potential moving target, human or otherwise. Here, the algorithm adopts the distance clustering procedure presented in [3]. It is based in the computation of the distance between two consecutive scan points, calculated by:

$$d(\mathbf{z}_i^*(k), \mathbf{z}_{i+1}^*(k)) = \|\mathbf{z}_{i+1}^*(k) - \mathbf{z}_i^*(k)\| = \sqrt{(r_{i+1}^*(k))^2 + (r_i^*(k))^2 - 2r_{i+1}^*(k)r_i^*(k)\cos(\varphi_{i+1}^*(k) - \varphi_i^*(k))} \quad (7)$$

If the distance given by (7) is less than a threshold chosen as:

$$d(\mathbf{z}_i^*(k), \mathbf{z}_{i+1}^*(k)) \leq D_0 + D_1 \min\{r_i^*(k), r_{i+1}^*(k)\} \quad (8)$$

where

$$D_1 = \sqrt{2(1 - \cos(\varphi_{i+1}^*(k) - \varphi_i^*(k)))} \quad (9)$$

the vector point  $\mathbf{z}_{i+1}^*(k)$  belongs to the same cluster as  $\mathbf{z}_i^*(k)$ .

The threshold is linear to the minimum distance between two consecutive scan points, due to the LMS200 beam divergence. The constant  $D_0$  allows an adjustment of the algorithm to noise and strong overlapping of pulses in close range. The linear  $D_1$  represents the lower bound associated with the laser's angular resolution.

Clusters have a minimum and maximum size of points established in the beginning (5 and 35 points respectively) rejecting that way possible outlier pairs and isolated points. Thus, it provides a very simple way of noise filtering. Clusters are represented as  $\mathbf{C}_{SEG}(k) = \{\mathbf{c}_1(k), \mathbf{c}_2(k), \dots, \mathbf{c}_q(k)\}$  with  $t = 1 \dots q$  and  $\mathbf{C}_{SEG}(k) \in \mathbf{Z}_{moving}^*(k)$  where each cluster can acquire different number of vector points and is represented as  $\mathbf{c}_i(k) = \{\mathbf{z}_1^*(k), \mathbf{z}_2^*(k), \dots, \mathbf{z}_p^*(k)\}$  with  $p \in \mathbf{N}$ .

Last but not least, a centre-of-gravity (COG) point-vector is maintained for every cluster by calculating the mean values of the vectors included in every  $\mathbf{c}_i(k)$ . Therefore,  $\bar{\mathbf{C}}_{SEG}(k) = \{\bar{\mathbf{c}}_1(k), \bar{\mathbf{c}}_2(k), \dots, \bar{\mathbf{c}}_q(k)\}$  corresponds to the centre-of-gravity vector.

## IV. TRACK INITIATIONS AND MAINTENANCE

### A. Track Initiation

A fundamental objective of any tracking system is to have one track number associated with each object under track. Having acquired specific moving targets from data clustering, the algorithm initiates new tracks for objects that are previously not assigned to any existing tracks. This is achieved by an enhanced two-point initialization process [11] utilizing a Munkres Optimal Assignment algorithm. Upon detection of a new moving object, which is described by cluster  $\mathbf{c}_i(k)$ , the cluster's COG point  $\bar{\mathbf{c}}_i(k)$  is set as object's initial dummy state vector.

$$\bar{\mathbf{x}}_{dum(DTr)}(k) = \begin{bmatrix} \bar{x}_{dum(DTr)}(k) \\ \bar{y}_{dum(DTr)}(k) \end{bmatrix} = \bar{\mathbf{c}}_i(k) \quad (10)$$

Derived from Batch Least Squares (LS) estimator formulation [11], we assume that the associated initial dummy error covariance of vector (10) is described by equation (4) as:

$$\mathbf{P}_{dum(DTr)}(k) = \bar{\mathbf{R}}_{lcart} \quad (11)$$

where  $DTr$  is the number of dummy tracks and  $DTr = 1 \dots f$ ,  $f \leq q$ . Therefore, the available dummy tracks at time  $k$  are represented as  $\bar{\mathbf{X}}_{dummy}(k) = [\bar{\mathbf{x}}_{Dum(1)}(k), \dots, \bar{\mathbf{x}}_{Dum(f)}(k)]$ , where  $\bar{\mathbf{X}}_{dummy}(k) \in \bar{\mathbf{C}}_{SEG}(k)$ .

Next, any available dummy tracks are validated and

promoted to verified-track status according to the following three-step procedure:

1. A *measurement innovation matrix* is calculated for every *DTr* based on the recursive LS estimator format [11]:

$$\mathbf{S}_{dum(DTr)}(k+1) = 2\bar{\mathbf{R}}_{lcart} \quad (12)$$

It is worth mentioning here that the time-invariant equation (12) is obtained by substituting (4), which is assumed as the noise covariance of all targets at every time step, and (11) into LS estimator's measurement innovation (Eq. 3.4.2-8 in [11]). For the same equation, measurement matrix  $\mathbf{H}$  is a unit matrix.

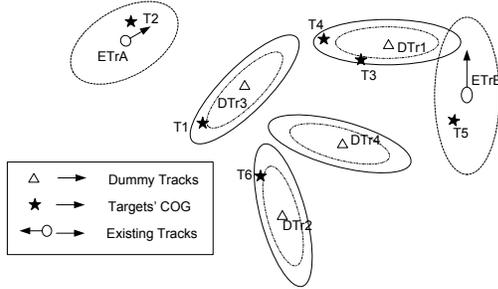


Fig. 4. A track initiation and maintenance scenario

2. A dummy track (*DTr*) is validated as a real track if and only if:

$$\gamma_0 < [\bar{\mathbf{c}}_i(k+1) - \bar{\mathbf{x}}_{dum(DTr)}(k)]^T [\mathbf{S}_{dum(DTr)}(k+1)]^{-1} [\bar{\mathbf{c}}_i(k+1) - \bar{\mathbf{x}}_{dum(DTr)}(k)] < \gamma_1, \quad t = 1 \dots q \quad (13)$$

where  $\gamma_0, \gamma_1$  are the appropriate chi-squared threshold levels [12], which determine a statistical validation zone.

Employing two validation thresholds instead of one provides additional moving-object filtering to the system. To relate moving-objects' speed to  $\gamma_0, \gamma_1$ , the following expressions were used. First the validation region volume for minimum and maximum speeds, which is centered on the initial measurement cluster's COG, is given by [13]:

$$V_{xy_{0/1}} = \left[ 2(\dot{x}_{min/max} T + 3\sqrt{\bar{R}_{lcart}^x}) \right] \times \left[ 2(\dot{y}_{min/max} T + 3\sqrt{\bar{R}_{lcart}^y}) \right] \quad (14)$$

where  $T$  is the sampling time interval and  $\dot{x}_{min/max}, \dot{y}_{min/max}$  are the minimum or maximum speeds in the  $X$  and  $Y$  directions respectively;  $\bar{R}_{lcart}^x, \bar{R}_{lcart}^y$  are variances of position measurements in these directions, obtained from the diagonal components of (4). In more general terms, the volume  $V$  of a validation region that corresponds to a threshold  $\gamma_{0/1}$  is:

$$V_{xy_{0/1}}(k) = c_{n_z} \gamma_{0/1}^{n_z/2} \sqrt{|S(k)|} \quad (15)$$

where  $n_z$  is the dimension of the measurement and  $c_{n_z}$  is the volume of the unit hypersphere of this dimension ( $c_1 = 2, c_2 = \pi, c_3 = 4\pi/3$ , etc.). By substituting (12) and (14) into (15) and solving with respect to  $\gamma_{0/1}$  we have:

$$\gamma_{0/1} = \frac{\left[ 2(\dot{x}_{min/max} T + 3\sqrt{\bar{R}_{lcart}^x}) \right] \times \left[ 2(\dot{y}_{min/max} T + 3\sqrt{\bar{R}_{lcart}^y}) \right]}{\pi \sqrt{|\mathbf{S}_{dum(DTr)}|}} \quad (16)$$

These time-invariant validation thresholds are the same for every detected target. Fig. 4 graphically demonstrates the validation procedure, where the solid ellipsoids around the dummy tracks represent the outer threshold  $\gamma_1$  and the dash-dotted ellipsoids correspond to inner threshold  $\gamma_0$ . In the same figure, stars symbolize targets' COG-points and the arrowed circles denote already existing tracks. When a target point lies in-between a dummy-track's  $\gamma_1$  and  $\gamma_0$ , a validation matrix (Fig. 5A) is utilized to assign the target-to-DTrack. For example, targets 1 and 6 are allocated to dummy tracks 3 and 2 respectively. A special case can be viewed in Fig. 4 where targets 3, 4 are associated with the same dummy track (*DTr1*). To solve conflicting situations such as this, a Munkres algorithm is applied to optimally assign the most probable target-to-DTrack pairings. Munkres is utilized only once per track, when a newly formed dummy-track needs to be associated with a measurement vector, at next time step. This is essential in order to acquire velocity rates and augment it to verified-track status. The method, which is similar to data assignment problem discussed in [14], is based on minimizing the normalized distance function (13) that has chi-square  $\chi_M^2$  distribution with  $M$  degrees of freedom, resulting in the optimally assigned validation matrix of Fig. 5A. Any dummy tracks without targets (*DTr4* in Fig. 4 and 5A) are simply deleted. Moreover, for any targets without dummy tracks (*T4*), a new dummy track is created. Targets associated to previously established tracks (Targets 2 and 5 to Tracks *ETrA* and *ETrB* respectively) are not taken into account.

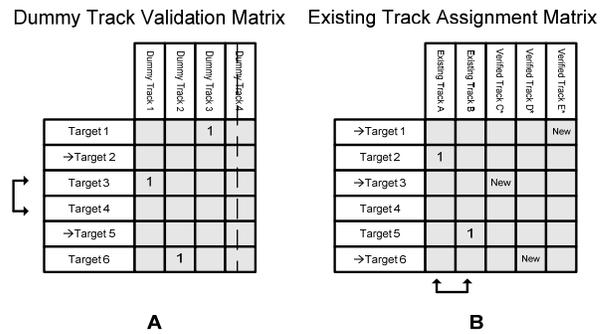


Fig. 5. Track initiation (A) and maintenance (B) matrices

3. The new verified tracks' state vectors assume the  $x$  and  $y$ -position components of  $\bar{\mathbf{c}}_i(k+1)$  and are augmented with velocity rates, computed from the difference of the two points and the elapsed time between them resulting in:  $\bar{\mathbf{x}}_{Ver(VTr)}(k+1) = [\bar{x}_{VTr}(k+1), \dot{x}_{VTr}(k+1), \bar{y}_{VTr}(k+1), \dot{y}_{VTr}(k+1)]^T$ . Initial

available tracks are represented by  $\bar{\mathbf{X}}_{\text{verified}}(k+1) = [\bar{\mathbf{x}}_{\text{ver}(1)}(k+1), \dots, \bar{\mathbf{x}}_{\text{ver}(s)}(k+1)]$  with  $VTr=1\dots s$ . A detailed explanation of the initial error covariance related to the new state vector is presented in [11].

### B. Track Maintenance and Termination

Before commencing new tracks for firstly-appeared measurement clusters, a JPDA algorithm (see section V for detailed description) generates assignment hypothesis for pairing any unassociated COG's in  $\bar{\mathbf{C}}_{\text{SEG}}(k)$  with the available existing tracks  $\hat{\mathbf{X}}_{\text{existing}}(k|k) = [\hat{\mathbf{x}}_1(k|k), \dots, \hat{\mathbf{x}}_a(k|k)]$ . However, maintaining track consistency requires a track-assignment matrix as the one displayed in Fig 5B. Continuing the scenario described in Fig. 4, the already established tracks *ETrA* and *ETrB* are assigned to targets 2 and 5. In the same matrix, any newly formulated tracks (eg. Track C, D, E) are started and placed at the end columns of the matrix for next-iteration use.

Verified tracks are divided into two categories: (a) Tracks with enhanced data-association performance, and (b) Tracks with reduced data-association performance. These two classes depend on how many association events have been accumulated by each track throughout its existing period. In other words, a linear counter assigned to every verified-track increases when the track is associated to a measurement cluster. Furthermore, as the track under examination remains unassociated, its counter decreases. Above a certain threshold, a track is considered as "enhanced" and it is regard as "more reliable" than tracks with less association hits than the threshold level.

Finally, tracks are considered obsolete and are terminated if they remain unassociated to measurement clusters for some specific period of time.

## V. THE TRACKING ALGORITHM

### A. Problem Formulation and Motion Models

Suppose that the trajectories of the available targets are described by  $\hat{\mathbf{X}}_{\text{existing}}(k|k)$  estimated vector. Now assume that the dynamics of each target can be modeled employing a number of  $n$ -hypothesized motion models denoted as  $M_n := \{1, 2, \dots, n\}$ . For the  $j^{\text{th}}$  hypothesized model ( $\mathbf{M}_r^j(k)$  active at time instant  $k$ ), the state dynamics and measurements of target  $r$  are modeled using the standard state equations:

$$\mathbf{x}_r(k+1) = \Phi_r^j(k)\mathbf{x}_r(k) + \mathbf{G}_r^j(k)\mathbf{w}_r^j(k), \quad j=1\dots n \quad (17)$$

$$\mathbf{z}_r(k+1) = \mathbf{H}_r^j(k+1)\mathbf{x}_r(k+1) + \mathbf{v}_r^j(k+1), \quad r=1\dots a \quad (18)$$

The state noise  $\mathbf{w}_r^j(k)$  and measurement noise  $\mathbf{v}_r^j(k+1)$  are assumed to be independent with each other, independent of the state components, white (uncorrelated in time), and Gaussian distributed with zero-mean and covariance

$$\mathbf{Q}_r^j(k) = E[\mathbf{w}_r^j(k)\mathbf{w}_r^j(k)^T] \quad \text{and} \quad \mathbf{R}_r^j(k+1) = E[\mathbf{v}_r^j(k+1)\mathbf{v}_r^j(k+1)^T] \quad \text{respectively.}$$

The system utilizes two linear dynamic motion models [11]. The Constant Velocity (CV) model with *state transition matrix*  $\Phi_{cv}(k)$  and *measurement noise transition matrix*  $\mathbf{G}_{cv}(k)$  as:

$$\Phi_{cv}(k) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{G}_{cv}(k) = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \quad (19)$$

And the Coordinated Turn (CT) model with constant angular velocity rate  $\omega$ :

$$\Phi_{ct}(k) = \begin{bmatrix} 1 & \frac{\sin \omega T}{\omega} & 0 & -\frac{1 - \cos \omega T}{\omega} \\ 0 & \cos \omega T & 0 & -\frac{\sin \omega T}{\omega} \\ 0 & \frac{1 - \cos \omega T}{\omega} & 1 & \frac{\sin \omega T}{\omega} \\ 0 & \sin \omega T & 0 & \cos \omega T \end{bmatrix} \quad \text{and} \quad \mathbf{G}_{ct}(k) = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \quad (20)$$

where  $T$  is the sampling time interval.

### B. JPDA-IMM

A brief description of the JPDA-IMM algorithm is provided in this section. A more detailed explanation can be found in [15], [16] and [17].

This algorithm mainly consists of six main steps. The following notations and definitions are used regarding the laser measurement vectors. The cumulative measurement set of validated measurements up to time  $k$  is denoted as  $\bar{\mathbf{Y}}(k) = \{\mathbf{Y}(1), \dots, \mathbf{Y}(k)\}$ . From among the unassociated measurements in  $\bar{\mathbf{C}}_{\text{SEG}}(k)$ , a set of validated measurements at time  $k$  is defined as  $\mathbf{Y}(k) = \{\mathbf{y}_1(k), \dots, \mathbf{y}_{\bar{q}}(k)\}$  were  $\mathbf{Y}(k) \in \bar{\mathbf{C}}_{\text{SEG}}(k)$  and  $\bar{q} \leq q$ .

**Step 1:** Interaction – mixing state, covariance and conditional mode probability from previous time instant. During this step the following quantities are calculated: (a) Predicted model probability:  $\hat{\mu}_r^j(k+1) = P\{\mathbf{M}_r^j(k+1) | \bar{\mathbf{Y}}(k)\}$ , (b) Mixing weight:  $\mu_r^{i/j}(k) = P\{\mathbf{M}_r^i(k) | \mathbf{M}_r^j(k+1), \bar{\mathbf{Y}}(k)\}$  for  $j, i \in M_n$ , (c) Mixing estimate:  $\bar{\mathbf{x}}_r^j(k|k) = E[\mathbf{x}_r(k) | \mathbf{M}_r^j(k+1), \bar{\mathbf{Y}}(k)]$ , (d) Mixing covariance:  $\bar{\mathbf{P}}_r^j(k|k)$ .

**Step 2:** State prediction  $\hat{\mathbf{x}}_r^j(k+1|k)$  and error covariance prediction  $\mathbf{P}_r^j(k+1|k)$ .

**Step 3:** Measurement validation procedure – For target  $r$ , the validation region is taken to be the same for all models, i.e. as the largest of them. Dominant model of target  $r$ :

$$j_r := \arg \left\{ \max_{j \in M_n} |\mathbf{S}_r^j(k+1)| \right\} \quad (21)$$

Then measurements in  $\bar{\mathbf{C}}_{\text{SEG}}(k)$  are validated if and only if:

$$[\mathbf{y}_r(k+1) - \hat{\mathbf{z}}_r^j(k+1)]^T [\mathbf{S}_r^j(k+1)]^{-1} [\mathbf{y}_r(k+1) - \hat{\mathbf{z}}_r^j(k+1)] < \gamma(k+1) \quad (22)$$

with  $t=1 \dots \bar{q}$ . Here, the standard algorithm presented in [15], [16], [17] was augmented by a time-variant validation threshold  $\gamma(k+1)$ , which is calculated by equation (16) for the residual covariance matrix  $\mathbf{S}_r^j(k+1)$ . This important and novel attribute allows the validation threshold to increase as measurement clusters are continuously associated to existing tracks, reaching a top value relative to objects' maximum speed and decrease inversely proportional to  $\mathbf{S}_r^j(k+1)$  when existing tracks remain unassociated to measurements. Using a time-variant validation threshold virtually eliminates conflicting situations where multiple tracks are assigned to the same measurement cluster.

**Step 4:** State estimation with validated measurements – All targets share a common validated measurement set  $\mathbf{Y}(k)$ . The JPDA [16] algorithm uses marginal association events  $\theta_r$ , which describe the hypothesis of a validated measurement  $\mathbf{y}_r(k)$  to be associated with (i.e. originates from) target  $r$ . Assuming that there are no unresolved measurements, a joint association event  $\Theta$  is effective when a set of marginal association events holds true simultaneously. That is  $\Theta = \bigcap_{r=1}^{\bar{q}} \theta_r$ . One can evaluate the likelihood that the target  $r$  is in model  $j_r$  as:

$$\Lambda_r^j(k+1) := \sum_{\Theta} p[\mathbf{Y}(k+1) | \Theta, \mathbf{M}_r^j(k+1), \bar{\mathbf{Y}}(k)] P\{\Theta\} \quad (23)$$

Furthermore, the probability of the marginal association event,  $\beta_r^{j_r}(k+1)$  as well as the Kalman gain  $\mathbf{K}_r^j(k+1)$ , the model-conditioned state estimation  $\hat{\mathbf{x}}_r^j(k+1|k+1)$  and the model-conditioned error covariance  $\mathbf{P}_r^j(k+1|k+1)$  are computed.

**Step 5:** Update of model probabilities:

$$\hat{\mu}_r^{j_r}(k+1) = P\{\mathbf{M}_r^j(k+1) | \bar{\mathbf{Y}}(k+1)\} = \frac{1}{c} \hat{\mu}_r^{j_r}(k+1) \Lambda_r^j(k+1) \quad (24)$$

where  $c$  is the normalization constant [15].

**Step 6:** Finally, global update of (a) state:  $\hat{\mathbf{x}}_r(k+1|k+1)$  and (b) error covariance matrix:  $\mathbf{P}_r(k+1|k+1)$  are obtained [17].

## VI. EXPERIMENTAL RESULTS

This section demonstrates the resulting performance of the 2D DATMO system based on the JPDA-IMM formulae. Several experiments have been performed to test this approach on a skid-steer drive PIONEER 3-AT mobile robot platform. It is equipped with a SICK LMS200 Laser Range Scanner, which delivers five measurement frames per second, resulting in a sampling interval of  $T = 200ms$  (Freq= 5Hz). When moving, the robot's maximum drive speed is  $0.3 m/sec$  and its maximum rotational velocity is  $10 deg/sec$ .

Potential targets (people) in robot environment move through various trajectories and speeds. It is assumed that an

average person walks at an average speed of  $1.5 m/sec$  [18], which set the maximum validation region for the target association procedure. Moreover, minimum and maximum speeds for computing the track-initiation thresholds ( $\gamma_0, \gamma_1$ ) are set to  $0.17 m/sec$  and  $0.5 m/sec$  respectively, for both  $X$  and  $Y$  directions.

To describe the objects' motion, the algorithm utilizes a CV model and two CT models, as described in section V., which operate in two counter-balancing angular velocity rates  $\omega = 0.8$  and  $\omega = -0.8 rad/sec$ . Due to state independent conditions, all process models are corrupted by white Gaussian-distributed noise  $\mathbf{w}_r^j(k)$ , with its components having a standard deviation of  $1.41422 m/sec^2$ . For the measurement noise model, described in section II., the following adjusted parameters were used:  $k_{\phi} = 0.000012$ ,  $k_{r_1} = 0.0007$  and  $k_{r_0} = 0.001$ . It is assumed that the detection probability [15] is  $P_d = 0.997$  and the transition probability matrix  $\pi_{\mu}$  is symmetric with uniformly distributed probabilities. The initial probability for each model is  $\hat{\mu}_r^{1..3}(0) = 0.333$ . Finally, the stationary-background attenuation threshold is  $\lambda = 1$ .

Due to space limitations in this paper, we present only a 4-second snapshot period (Fig. 6) from a tracking scenario that lasted several minutes (16min total time). During this experiment the robot traveled inside a  $10m \times 6m$  lab-space environment through a number of different rotational and translational trajectories. At the same time four people were moving freely inside the lab. By observing Fig. 6, it is clearly visible that four tracks were initiated and associated to measurements acquired by the laser scanner through the snapshot period. The black dots represent the sequential COG points obtained from measurement clustering as time progresses and the circles/squares represent the estimated trajectories (tracks) of the detected moving objects. These trajectories are grouped together under different-lined (dashed, dashed double-dotted, solid, dotted) elliptic shapes in order to distinguish the four different initiated tracks. Furthermore, in the last estimated location of each track, a heading arrow is attached to indicate the direction of moving target. The recursively estimated track-points assume a different shape (from circles to squares) after they have continuously been associated with measurements for a period of 2 seconds (10 sequential data associations) in order to designate a track as "enhanced" (see section IV, B).

Finally, tracks 3 and 4 are not initiated until the end of the 4-second snapshot period as shown in Fig. 6. This is due to the fact that in order to establish a track, the moving object must maintain a speed between  $0.17 m/sec$  and  $0.5 m/sec$  for more than one system iterations. Although this can result in some excess track-initiation delays, increasing the gap between minimum and maximum track threshold speeds could introduce additional unwanted tracks due to appearance of some outlier background points that could not be

eliminated in background attenuation and clustering procedures. However, the selected speed thresholds provide adequate speed limitations to initiate a correct track within a 4 sec period and minimize potential risks in commencing erroneous tracks.

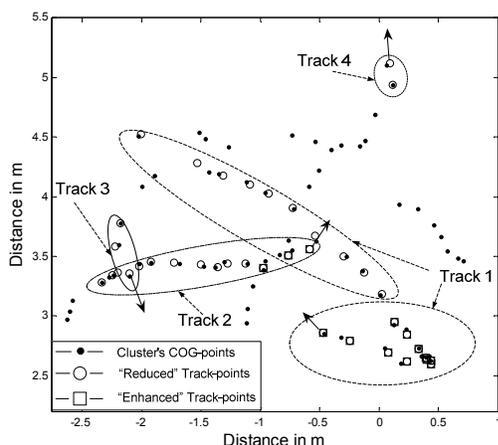


Fig. 6. Four tracks initiated and associated to four detected moving targets during a 4sec-period (20 algorithm iterations).

## VII. CONCLUSIONS

All collected results indicate that the proposed DTMO framework is described by the following attributes: (a) It successfully detects and tracks multiple moving targets in dynamic environments providing at the same time stationary-background attenuation from a moving observer as well as a high trajectory-estimation accuracy (about 15cm maximum real-to-estimated position error obtained from Fig. 6). (b) The system effectively distinguishes targets when move at close proximity to each other. It assigns and maintains only one track per target at all times and terminates unnecessary tracks. (c) The number of simultaneous targets under track is limited only by hardware recourses, thus it could, theoretically speaking, track an infinite number of moving objects. (d) The system offers improved computational and memory characteristics since it only require a single occupancy grid-map to attenuate the environmental stationary background information.

As future work, it is in the purpose of this research to examine alternative map-building techniques where the linear thresholds described in section II, *C* will be replaced by its non-linear counterparts. Furthermore, kd-trees will provide an increased throughput during the dynamic background attenuation procedure (section II, *D*).

## REFERENCES

[1] H. Zhao and R. Shibasaki, "A Novel System for Tracking Pedestrians Using Multiple Single-Row Laser-Range Scanners." In *IEEE Trans. On Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 2, March 2005, pp. 283-291.

[2] J. Xavier, M. Pacheco, D. Castro, A. Ruano and U. Nunes, "Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver." In *Proceedings of 2005 IEEE International Conference on Robotics and Automation (ICRA'05)*, Barcelona, Spain, 18-22 April 2005, pp. 3930-3935.

[3] D. Castro, U. Nunes and A. Ruano, "Feature Extraction and for Moving Objects Tracking System in Indoor Environments." In *Proceedings of 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles (IAV'04)*, Lisbon, 5-7 July 2004.

[4] K. O. Arras, O. M. Mozos and W Burgard, "Using Boosted Features for the Detection of People in 2D Range Data." In *Proceedings of 2007 IEEE International Conference on Robotics and Automation (ICRA'07)*, Roma, Italy, 10-14 April 2007, pp 3402-3407.

[5] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking Multiple Moving Objects With a Mobile Robot using Sample-Based Joint Probabilistic Data Association Filters." In *International Journal of Robotics Research (IJRR)*, vol. 22, no. 2, 2003, pp. 99-116.

[6] J. Almeida, A. Almeida and R. Araujo, "Tracking Multiple Moving Objects for Mobile Robotics Navigation." In *Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'05)*, Catania, Italy, 19-22 September 2005, pp. 203-210.

[7] M. Lindstrom and J.-O. Eklundh, "Detecting and Tracking Objects from a Mobile Platform using a Laser Range Scanner." In *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*, Wailea, Hawaii, October/November 2001, Vol 3, pp.1364 – 1369.

[8] C. Wang, C. Thorpe, and S. Thrun, "Online Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas." In *Proceedings of 2003 IEEE International Conference on Robotics and Automation (ICRA'03)*, vol. 1, Sept. 2003, pp. 842–849.

[9] H. Baltzakis and P. Trahanias, "Using Multi-hypothesis Mapping to Close Loops in Complex Cyclic Environments." In *Proceedings of 2001 IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, Florida, 15-19 May 2006, pp. 824-829.

[10] F. Lu and E E Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans." In *Proceedings 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994, pp. 935-938.

[11] Y. Bar-Shalom, X. Rong Li and T. Kirubarajan, "Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software." John Wiley & Sons, Inc, 2001.

[12] Y. Bar-Shalom and X. Rong Li, "Multitarget-Multisensor Tracking: Principles and Techniques." Storrs, CT: YBS Publishing, 1995.

[13] T. Kirubarajan and Y. Bar-Shalom, "Probabilistic Data Association Techniques for Target Tracking in Clutter." In *Proceedings of the IEEE*, vol 92, Issue 3, Mar 2004, pp. 536 - 557

[14] P. Konstantinova, M. Nikolov, T. Semerdjiev, "A Study of Clustering Applied to Multiple Target Tracking Algorithm." In *Proceedings of 5th International Conference on Computer Systems and Technologies (CompSysTech'04)*, Rousse, Bulgaria, June 2004, pp.IIIA.22-1-III.A.22-6.

[15] Y. Bar-Shalom and W. D. Blair, "Multitarget-Multisensor Tracking: Applications and Advantages, Volume III." Artech House Publishers Boston – London, 2000.

[16] B. Chen and J. K. Tugnait, "Tracking of Multiple Maneuvering Targets in Clutter Using IMM/JPDA Filtering and Fixed-lag Smoothing." In *Journal of Automatica*, vol. 37, no. 2, February 2001, pp. 239-249.

[17] P. Kondaxakis, S. Kasderidis, P. Trahanias, "Tracking Multiple Targets from a Mobile Robot Platform using a Laser Range Scanner." In *Proc. IET on Target tracking and data fusion: Algorithms and applications*, Birmingham, UK, 15 -16 April 2008, pp. 177-184.

[18] URL: [http://en.wikipedia.org/wiki/Orders\\_of\\_magnitude\\_\(speed\)](http://en.wikipedia.org/wiki/Orders_of_magnitude_(speed))